COMP9414 ASS2

YIZHENG YING
Z5141180

Q1:

a)

	Start10	Start12	Start20	Start30	Start40
UCS	2565	Mem	Mem	Mem	Mem
IDS	2407	13812	5297410	Time	Time
A*	33	26	915	Mem	Mem
IDA*	29	21	952	17297	112571

b)

For the UCS, its usage of memory is the largest of all these four algorithms and when the start number becomes bigger, it will run out of memory;

For the IDS, it is a kind of Depth-Limited Search, so the space it takes is only O(bk) and it is the most memorial efficient of all the algorithms, but it takes a large amount of time to get the results, sometimes we even do not konw how long time it will take to get the results;

For the A*, the time to get results is faster than others, but sometimes it will also run out of memory when the start number get larger;

For the IDA*, it seems the most efficient of all algorithms both in time and memory and it is the only one who can get the result of Start40.

Q2:

a)

	Start50		Start60		Start64	
Greedy	164	5447	166	1617	184	2174

b)

	Start50		Start60		Start64	
1.2	52	191438	62	230861	66	431033

```
% Otherwise, use Prolog backtracking to explore all successors
% of the current node, in the order returned by s.
% Keep searching until goal is found, or F_limit is exceeded.
depthlim(Path, Node, G, F_limit, Sol, G2)  :-
    nb_getval(counter, N),
    N1 is N + 1,
    nb_setval(counter, N1),
    % write(Node),nl,    % print nodes as they are expanded
    s(Node, Node1, C),
    not(member(Node1, Path)),      % Prevent a cycle
    G1 is G + C,
    h(Node1, H1),
    W is 1.2,
    F1 is (2-W)*G1 + W*H1,
    F1 =< F_limit,
    depthlim([Node|Path], Node1, G1, F_limit, Sol, G2).
```

c)

	Start50		Start60		Start64	
1.4	66	116174	82	3673	94	188917
1.6	100	34647	148	55626	162	235852

d)

When the w is incresing from 1.2 to 1.6, the number of G is getting closer

to the number of G when we use the Greedy and the number of N is getting smaller than before.The speed of solution is becoming faster when w is increasing but the quality of solution is getting fewer.

Q3:

a)

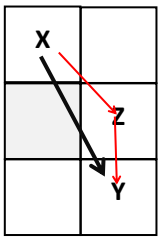
Using the Manhattan Distance heuristic to dominate the Straight-Line-Distance heuristic, the formula for it in the format is:

$$h(x, y, xG, yG) = |x-xG|+|y-yG|$$

b)

(i)

The Straight-Line-Distance heuristic is not admissible, for example, from X to Y, using SLD heuristic , we need to follow the red arrows from X to Z then to Y, the cost will be 2, but actually we can follow the black arrow and the cost will be $\sqrt{5}$ which is larger than 2, beacuse h() is called admissible if $\forall n\, h(n) \leq h^*(n)$ where $h^*(n)$ is true cost from n to goal, but in this case, true cost is 2 which is smaller than h(n), so the SLD heuristic is not admissible.



(ii)

No, it is not admissible.Because the costs of moving vertially, diagonally, horizontally are the same, but in part(a), the cost will be the sum of the

costs of moving vertially and horizontally, it must be larger than just moving vertially or just moving horizontally.So it is not admissible.

(iii)

When $|x-xG| \geq |y-yG|$:

$$h(x, y, xG, yG) = |x-xG|$$

When $|x-xG| < |y-yG|$:

$$h(x, y, xG, yG) = |y-yG|$$

Q4:

a)

n	Time	Optimal sequence
0	**0**	
1	**2**	**+ -**
2	**3**	**+ 0 -**
3	**4**	**+ 0 0 -**
4	**4**	**+ + - -**
5	**5**	**+ + - 0 -**
6	**5**	**+ + 0 - -**
7	**6**	**+ + 0 - 0 -**
8	**6**	**+ + 0 0 - -**
9	**6**	**+ + + - - -**
10	**7**	**+ + + - - 0 -**
11	**7**	**+ + + - 0 - -**
12	**7**	**+ + + 0 - - -**
13	**8**	**+ + + 0 - - 0 -**
14	**8**	**+ + + 0 - 0 - -**
15	**8**	**+ + + 0 0 - - -**
16	**8**	**+ + + + - - - -**
17	**9**	**+ + + + - - - 0 -**
18	**9**	**+ + + + - - 0 - -**
19	**9**	**+ + + + - 0 - - -**
20	**9**	**+ + + + 0 - - - -**

21	**10**	**+ + + + 0 - - - 0 -**

b)

n	Time	Number of '+'	Number of	2s+1	2s+2
0		**0**	**0**		
1	**2**	**1**	**0**		**2**
2	**3**	**1**	**1**	**3**	
3	**4**	**1**	**2**		**4**
4	**4**	**2**	**0**		**4**
5	**5**	**2**	**1**	**5**	
6	**5**	**2**	**1**	**5**	
7	**6**	**2**	**2**		**6**
8	**6**	**2**	**2**		**6**
9	**6**	**3**	**0**		**6**
10	**7**	**3**	**1**	**7**	
11	**7**	**3**	**1**	**7**	
12	**7**	**3**	**1**	**7**	
13	**8**	**3**	**2**		**8**
14	**8**	**3**	**2**		**8**
15	**8**	**3**	**2**		**8**
16	**8**	**4**	**0**		**8**
17	**9**	**4**	**1**	**9**	
18	**9**	**4**	**1**	**9**	
19	**9**	**4**	**1**	**9**	
20	**9**	**4**	**1**	**9**	
21	**10**	**4**	**2**		**10**

From the queation, we know that:

$$\lceil 2\sqrt{n} \rceil = \begin{cases} 2s+1, & \text{if } \quad s^2 \quad < n \leq s(s+1) \\ 2s+2, & \text{if } s(s+1) < n \leq \ (s+1)^2 \end{cases}$$

s represents the time of acceleration, because the time of slowing down is the same as the time of acceleration, so ,generally the total time is 2s, but in some situations, we need to keep the speed for one or two times. From

the table above we can see that the results of 2s+1 or 2s+2 are same as the results of Time.As we know that M(n,0) equals Time, so:

$$M(n,0) = \lceil 2\sqrt{n} \rceil$$

c)

We assmue that |a|=1,so the total time is t, the time of acceleration from 0 to k is k.When M(n,k), the time of acceleration is t1, the time of slowing down is t2,so:

$$t=t1+t2+k$$

Because t=M(n1,0),so t1+t2=M(n1,0)-k.

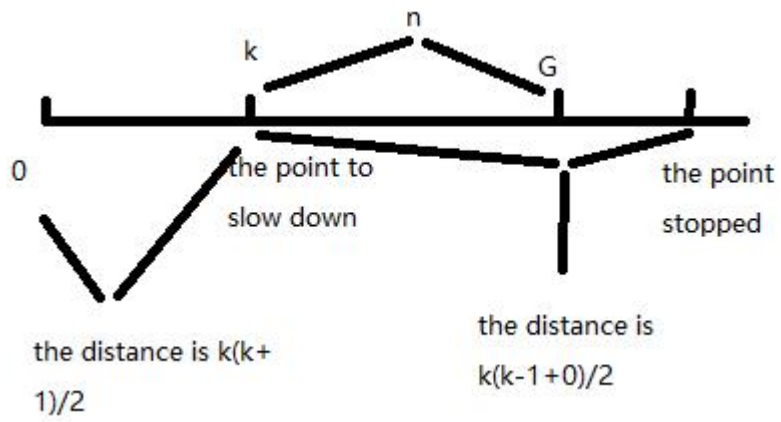
The distance of acceleration from 0 to k is n2,from the Gaussian summation formula:

$$n2=(1+k)*k/2$$

As n1=n+n2,so:

$$M(n,k) = \left\lceil 2\sqrt{n+\frac{1}{2}(k+1)k} \right\rceil - k$$

d)



Like the picture above, the distance 0 point to stop point is k(k+1)/2+k(k-1)/2, so the distance from stop point back to G point is k(k-1)/2-n, from the question c we can know that:

$$M(n,k)=\left\lceil 2\sqrt{\frac{1}{2}k(k+1)+\frac{1}{2}k(k-1)}\right\rceil+\left\lceil 2\sqrt{\frac{1}{2}k(k-1)-n}\right\rceil-k=\left\lceil 2\sqrt{\frac{1}{2}k(k-1)-n}\right\rceil+\mathrm{k}$$

e)

$$h(r,c,u,v,rG,cG) = \max(M(rG-r,u),\ M(cG-c,v))$$